REMARKS

By this amendment, claims 5 and 6 have been cancelled and claims 1-3 and 7 have been amended. Currently, claims 1-4 and 7-29 are pending in the application.

Claims 1-29 were rejected under 35 USC 112, second paragraph, as being indefinite. The Examiner believed that the claims did not include a combinational feature showing how the modules are employed to result in the intended invention. This rejection is respectfully traversed in view of the amendments made to the claims and the remarks below.

The present invention includes a system for dynamically generating and processing a program. This system operates so that when a plurality of definition information for defining the processing are inputted in a client pc, the plurality of definition information are sent to the server pc. In the server pc, a plurality of configuration information are read—out according to the definition information and then a plurality of functional module classes are read—out according to the request included in the configuration information. The plurality of

functional module classes are sent to the client pc and then the unit-program generating is generated by using the coded processing logic of plurality of functional module classes and this is executed by using the processing condition included in the configuration information. Claims 1 and 3 have been amended to recite the locations of the various elements in the claims.

The present invention also relates to a client computer and a server computer in the system, a method for dynamically generating and processing a program, a computer-readable and -recordable media for controlling a server computer comprising a system for dynamically generating and processing a program as well as a program transfer system comprising a system for dynamically generating and processing a program. The elements of all of these claims are believed to be definite and clear and set forth the features of the present invention. Accordingly, it is respectfully requested that this rejection be withdrawn in view of these amendments and comments. Should the Examiner believe that the claims should be amended to include further limitations to overcome this rejection, the Examiner is respectfully

requested to give applicant a suggestion for overcoming this rejection.

Claims 1-29 were rejected under 35 USC 103(a) as being obvious over Yokote (U.S. Patent No. 6,105,074). The Examiner believed that Yokote teaches a functional module storage means (col. 13, lines 64, col. 14, lines 3, col. 1, lines 14-20), a configuration information storage means (col. 18, lines 33-37), a definition information input means (col. 10, lines 45-54), a configuration information read—out means (col. 18, lines 33-37), a unit-program generating means (col. 2, lines 13-20 & col. 17, lines 13-21) and a unit—program processing means (col. 17, lines 13-21).

As discussed above, the present invention includes a system for dynamically generating and processing a program. This system operates so that when a plurality of definition information for defining the processing are inputted in a client pc, the plurality of definition information are sent to the server pc. In the server pc, a plurality of configuration information are read-out according to the definition information and then a plurality of functional module classes are read-out according to

the request included in the configuration information. The plurality of functional module classes are sent to the client pc and then the unit-program generating is generated by using the coded processing logic of plurality of functional module classes and this is executed by using the processing condition included in the configuration information.

Yokote discloses that when downloading the application program from the sever pc to the client pc and executing the application program, the server pc checks the execution environment for executing the application program in the client pc. If the client pc has an execution environment that is able to execute the application program, the server pc downloads the application program to the client pc. If not, the server pc downloads the objects for generating the execution environment for executing the application program in the client pc and then downloads the application program to the client pc.

As a result, it is possible to execute the application program in the client pc independent of the operating software in the client pc.

Also Yokote discloses that in the application program a

plurality of objects are executed concurrently with sending and receiving messages from each other. This is generally employed in an Object Oriented program such as JAVA program, C^{++} program and Visual BASIC program.

In the present invention, a plurality of definition information for defining the processing are needed to be input in the client pc. See Figs. 7 - 17.

In Yokote, it is only required to select the application program by using the client pc. See col. 11, line 1 - when a user selects a movie. -

That is, according to the present invention, it is possible to generate the desired voluntary program by using the client pc even if the user has no knowledge of the programming language. But in Yokote, the user does not generate the application program by using the client pc and the application program that is pre-generated by programmer is downloaded and executed in the client pc.

Further, the Examiner believed that the functional module classes of the present invention is similar to the object of Yokote, however, applicant respectfully submits that this is

misunderstanding by the Examiner.

Applicant believes that it seems that the object of Yokote equates with a unit-program of the present invention. See col. 3, line 2: the objects are defined as concurrent objects and col. 3, lines 46-47: the application program 11 comprises a plurality of objects 14 gathered together.

The functional module classes of the present invention have a coded processing logic and a unit—program is generated by using a plurality of functional module classes.

Accordingly applicant respectfully submits that the functional module classes of the present invention differ from the object of Yokote. Specifically, Yokote do not teach or suggest that the object comprises the functional module classes of the claimed invention.

Further, the Examiner stated that Yokote teaches the definition information input means (col. 10, lines 45-54). However, reviewing col. 10, lines 45-54 of Yokote, the device driver is disclosed that drives the device such as the keyboard, display, printer, or HDD according to the application program,

and the device driver is a program and differs from the definition information.

That is, the definition information in the present invention is written by the non-programming language (see Figs. 1 - 17), however, in Yokote, the device driver is written by the programming language. Accordingly, Yokote does not teach the definition information input means.

Also, the Examiner stated that Yokote teaches the configuration information storage means and the configuration information read–out means in col. 18, lines 33-37.

However, according to col. 18, lines 33-37 of Yokote, this section discloses that when the transferred object and the object in the client pc are inconsistent, the consistent object is downloaded to the client pc. That is, Yokote does not teach or suggest the configuration information that includes at least the request information and the condition for executing the unit-program.

Additionally, the Examiner stated that Yokote teaches the unit-program generating means in col. 2, lines 13-20 & col. 17, lines 13-21. However, according to col. 2, lines 13-20 & col.

17, lines 13-21 of Yokote, this section teaches to compile or interpret the downloaded object into binary code that the CPU can process, and to compile or interpret the downloaded object to the intermediate code and generate the binary code by using the intermediate code. In other words, Yokote uses a word 'generate' but the intermediate code in Yokote is the compiled or interpreted object and also equates with the binary code. Thus the intermediate code differs from the functional module classes since the plurality of functional module classes are used for generating the unit-program. Yokote never teaches the features of the unit-program generating means.

The various claim features discussed above are selectively found in the other independent claims. For this reason and the reasons discussed above, applicant respectfully submits that the independent claims are allowable over Yokote.

Further, claims 18-20 relate to a method for dynamically generating and processing a program that are executed by the system similar to claims 1-17. Claims 21-25 relate to a computer—readable media recorded the program for controlling the server pc and client pc according to the system similar to claims

1-17. Claims 26-29 relate to a program transfer system for transferring and downloading the program wherein the program is the controlling program for controlling the server pc and client pc similar to claims 1-17.

Therefore, in view of foregoing amendments and remarks, it is respectfully submitted that claims 1-4 and 7-29 are allowable over the prior art of record. Thus, applicant respectfully submits that the application is now in condition for allowance and an action to this effect is respectfully requested.

If there are any questions or concerns regarding the amendments or these remarks, the Examiner is requested to telephone the undersigned at the telephone number listed below.

Respectfully submitted,

Date: June 13, 2005

Randolph A. Smith
Reg. No. 32,548

SMITH PATENT OFFICE
1901 Pennsylvania Ave., N.W.
Suite 200
Washington, DC 20006-3433
Telephone: 202/530-5900
Facsimile: 202/530-5902
Shimoji061305